

6 said program converting unit generates a machine language instruction from a source
E7 program for a processor out of an embedded-type custom processor series which has an address
8 width N in accordance with a necessary program size, and
9

said processor comprises:

10 memory means for storing a program, the memory means having a minimum storage
11 capacity of 2^N bytes to store the program and having N address lines, the program including an
12 N-bit data arithmetic operation instruction and other instructions operating on both N-bit and
13 M-bit data, N being greater than M; and

14 a processor core having an address bus of N bits which is equal in size to the number of
15 address lines of the memory means, the processor core being selected from a plurality of
16 processor cores,

17 wherein the processor core includes:

18 a program counter for holding an N-bit instruction address to output an instruction at the
19 N-bit address to the memory means, the program counter having an N-bit address length which
20 is equal in size to the number of address lines of the memory means;

21 fetching means for fetching an instruction from the memory means using an N-bit
22 instruction address from said program counter; and

23 executing means for executing all N-bit arithmetic operation instructions and for
24 executing other instructions except for arithmetic operation instructions at one of N-bit length
25 and M-bit length, the executing means having N-bit length,

26 whereby an N-bit address is calculated by the N-bit arithmetic operation independently
27 of a data bit-width, said data bit-width being M, and

28 said program converting unit comprises:

29 parameter holding means for holding a data width M and a pointer width N, said data
30 width M representing the number of bits of data used in the source program, said pointer width
31 N representing the number of bits of an address, said N and M being input by a user in
32 accordance with program size; and

33 generating means for generating an instruction based on the source program to set the
34 data width M as valid when a variable used in a machine language instruction to be generated
35 is a variable showing data, and for generating an instruction to set the address width N as valid
36 when a variable used in a machine language instruction to be generated is a variable representing
37 an address,

38 wherein the program converting unit generates a unique set of machine language
39 instructions from the source program for each N specified by the user.

50
51
1 ~~59~~. The computer system of Claim ~~58~~⁴⁹, wherein the processor further comprises:
2 an address register group including a plurality of N-bit address registers;
3 a data register group including a plurality of N-bit data registers,
4 wherein said executing means executes the N-bit and M-bit data operation instructions
5 using the address registers, while executing the M-bit data operation instruction using data
6 registers.

51
1 ~~60~~. The computer system of Claim ~~59~~⁵⁰, wherein:
2 said N is 24 and said M is 16; and
3 said processor is installed in a 1-chip microcomputer, whereby said 1-chip microcomputer
4 becomes suitable for running a program that utilizes a memory over 64 Kbytes for an operation
5 with 16-bit data.

52
1 ~~61~~. The computer system of Claim ~~60~~⁵¹, wherein the processor further comprises:
2 compensating means for extending an effective bit-width of the data in one of the address
3 registers and the data register to 24 bits,
4 wherein said compensating means operates in accordance with a compensate instruction
5 entered after a machine language instruction designating an arithmetic operation that will
6 possibly cause an overflow.

53

52

62

The computer system of Claim ~~61~~⁵², wherein said compensating means includes:

a first extending unit for filling a logical value of a sign bit in all bits higher than the effective bit-width in a register; and

a second extending unit for filling a logical value "0" in all bits higher than the effective bit-width in a register.

56

49

63

The computer system of Claim ~~58~~⁴⁹, wherein the processor further comprises:

an address register group including a plurality of N-bit address registers; and

a data register group including a plurality of M-bit data registers,

wherein said executing means executes one of an N-bit data operation instruction and an M-bit data operation instruction using the address registers, while executing the M-bit data operation instruction using the data registers.

57

56

64

The computer system of Claim ~~63~~⁵⁶, wherein:

said N is 24 and said M is 16; and

said processor is installed in a 1-chip microcomputer, whereby said 1-chip microcomputer becomes suitable for running a program that utilizes a memory over 64 Kbytes for an operation with 16-bit data.

58

57

65

The computer system of Claim ~~64~~⁵⁷, wherein the processor further comprises:

compensating means for extending an effective bit-width of the data in one of the address registers and the data register to 24 bits,

wherein said compensating means operates in accordance with a compensate instruction entered after a machine language instruction designating an arithmetic operation that will possibly cause an overflow.

59

58

1 ~~66.~~ The computer system of Claim ~~65~~, wherein said compensating means includes:
 2 a first extending unit for filling a logical value of a sign bit in all bits higher than the
 3 effective bit-width in a register;
 4 a second extending unit for filling a logical value "0" in all bits higher than the effective
 5 bit-width in a register.

54

56

1 ~~67.~~ The computer system of Claim ~~59~~, wherein the processor further comprises:
 2 compensating means for extending an effective bit-width of the data in one of the address
 3 registers and the data register to N bits,
 4 wherein said compensating means operates in accordance with a compensate instruction
 5 entered after a machine language instruction designating an arithmetic operation that will
 6 possibly cause an overflow.

55

54

1 ~~68.~~ The computer system of Claim ~~67~~, wherein said compensating means includes:
 2 a first extending unit for filling a logical value of a sign bit in all bits higher than the
 3 effective bit-width in a register; and
 4 a second extending unit for filling a logical value "0" in all bits higher than the effective
 5 bit-width in a register.

60

49

1 ~~69.~~ The computer system of Claim ~~58~~ wherein the pointer width N and the data width
 2 M are input by a user during an execution of the program converting unit.

148

61
70. 1 A computer system comprising a central processing unit and a software program
2 compiler, wherein

3 the central processing unit is one of a series of processing units, each processing unit
4 having a different address length N, N being longer than a data width M, the address length of
5 the processing unit selected based on a size of a source program, the processing unit comprising:

6 memory means for storing a program, the memory means having a minimum storage
7 capacity of 2^N bytes to store the program and having N address lines, the program including an
8 N-bit data arithmetic operation instruction and other instructions operating on both N-bit and
9 M-bit data, N being greater than M; and

10 a processor core having an address bus of N bits which is equal in size to the number of
11 address lines of the memory means, the processor core being selected from a plurality of
12 processor cores,

13 wherein the processor core includes:

14 a program counter for holding an N-bit instruction address to output an instruction at the
15 N-bit address to the memory means, the program counter having an N-bit address length which
16 is equal in size to the number of address lines of the memory means;

17 fetching means for fetching an instruction from the memory means using an N-bit
18 instruction address from said program counter; and

19 executing means for executing all N-bit arithmetic operation instructions and for
20 executing other instructions except for arithmetic operation instructions at one of N-bit length
21 and M-bit length, the executing means having N-bit length,

22 compensating means for extending an effective bit-width of the data in one of the address
23 registers and the data register to N bits, wherein the compensating means compensates as
24 directed by a compensate instruction which is entered after a machine language arithmetic
25 instruction which may cause an overflow;

26 whereby an N-bit address is calculated by the N-bit arithmetic operation independently
27 of a data bit-width, said data bit-width being M, and

28 the software compiler comprises:

29 parameter holding means for holding a data width M and a pointer width N, the data

30 width M representing the number of bits of data used in the source program, the pointer width

31 N representing the number of bits of an address, N and M being inputs to the compiler input by

32 a user during an execution of the compiler, N and M selected by the user based on the size of

33 the source program; and

34 generating means for generating an instruction based on the source program to set the

35 data width M as valid when a variable used in a machine language instruction to be generated

36 is a variable showing data, and for generating an instruction to set the address width N as valid

37 when a variable used in a machine language instruction to be generated is a variable representing

38 an address,

39 wherein the program converting unit generates a unique set of machine language

40 instructions from the source program for each N specified by the user.